IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/870,614	Confirmation No.	1779
Applicant:		Scott J. Broussard		
Title:		Dynamic Buffering of Graphic Images by a Platform Independent Application Program Interface		
Filed	:	May 31, 2001		
TC/A.U.	:	2173		
Examiner	:	Dennis G. Bonshock		
Docket No.	:	AUS920010265US1		
Customer No.	:	48,916		

BRIEF ON APPEAL

Mail Stop Appeal Briefs-Patents
Commissioner for Patents
P.O. Box 1450, Alexandria, VA 22313-1450

This is an appeal from the Final Office Action, mailed on October 5, 2009 (paper no. 20090930), finally rejecting claims 1-22. A Notice of Appeal was filed on January 5, 2010.

The fee for filing this Brief on Appeal is $540.00 and is being paid electronically at the time of filing of this Brief. If this amount is insufficient, or should any additional fees under 37 C.F.R. § 1.16 to 1.21 be required for any reason relating to the enclosed materials, the Commissioner is authorized to deduct said fees from IBM Corporation, Deposit Account No. 09-0447.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corp. (IBM) of Armonk, New York

RELATED APPEALS AND INTERFERENCES

The current Application was the subject of an Appeal (No. 2007-3520) in which the Board of Patent Appeals and Interferences (BPAI) rendered an opinion on March 31, 2008, reversing the Examiner's rejections of claims 1-22.

In addition, Appeals have been filed in the following related applications:

09/870,613: Notice of Appeal filed 02/07/2005; Appeal Brief filed 04/07/2005; Examiner's Answer mailed 08/25/2005; Supplemental Appeal Brief filed 04/24/2006; Examiner's Answer mailed 07/17/2006; Appeal Brief filed 05/22/2007; Examiner's Answer mailed 08/28/2007; BPAI Decision – Examiner Affirmed in Part 11/24/2008 (Appeal No. 2008-2534); Notice of Allowance mailed 03/10/2009; U.S. Patent No. 7,562,306 issued 07/14/2009.

09/870,615: Notice of Appeal file 09/14/2004; Appeal Brief filed 11/09/2004; Notice of Allowance mailed 03/03/2005; U.S. Patent No. 6,918,093 issued 07/12/2005.

09/870,620: Notice of Appeal filed 12/07/2004; Appeal Brief filed 02/11/2005; Examiner's Answer mailed 05/20/2005; Appeal Brief filed 04/20/2006; Examiner's Answer mailed 06/28/2006; Appeal Brief filed 11/22/2006; Examiner's Answer mailed 04/06/2007; BPAI Decision – Examiner Affirmed in Part 03/31/2008 (Appeal No. 2008-0155); RCE filed 10/30/2008; Notice of Allowance mailed 03/20/2009; U.S. Patent No. 7,571,388 issued 08/04/2009.

09/870,621: Notice of Appeal filed 09/24/2004; Appeal Brief filed 11/23/2004; Examiner's Answer mailed 04/05/2005; Appeal Brief filed 11/15/2006; Examiner's Answer mailed 03/07/2007; BPAI Decision – Examiner Reversed 03/31/2008 (Appeal No. 2007-2696); Notice of Allowance mailed 08/04/2008; RCE filed 10/30/2008; Notice of Allowance mailed 03/17/2009; U.S. Patent No. 7,571,389 issued 08/04/2009.

09/870,622: Notice of Appeal file 08/24/2004; Appeal Brief filed 11/01/2004; Examiner's Answer mailed 04/05/2005; Appeal Brief filed 08/03/2006; Appeal Brief filed 11/07/2006; Examiner's Answer mailed 01/29/2007; BPAI Decision – Examiner Reversed 03/31/2008 (Appeal No. 2008-0098); Notice of Allowance mailed 07/31/2008; RCE filed 10/30/2008.

09/870,624: Notice of Appeal filed 05/23/2005; Appeal Brief filed 07/25/2005; Examiner's Answer mailed 10/07/2005; Appeal Brief filed 04/20/2006; Examiner's Answer mailed 07/17/2006; Appeal Brief filed 11/22/2006; Examiner's Answer mailed 03/07/2007; BPAI Decision – Examiner Affirmed in Part 03/31/2008 (Appeal No. 2008-0463).

The Appellants are not aware of any other related appeals, interferences or judicial proceedings that will directly affect, be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 1-22 are currently pending and have been finally rejected.

Claims 1, 2, 5, 6, 20 and 21 are rejected under 35 U.S.C. §103(a) as being unpatentable over DiNicola et al. (U.S. Pat. No. 4,951,229; hereinafter referred to as "DiNicola") in view of Ross (U.S. Pat. No. 5,8838,336) and further in view of Nagata (U.S. Pat. No. 6,522,341).

Claims 3, 4 and 7-10 are rejected under 35 U.S.C. §103(a) as being unpatentable over DiNicola, Ross and Nagata and further in view of Fowler ("Mixing Heavy and Light Components").

Claims 11-13, 18, 19 and 22 are rejected under 35 U.S.C. §103(a) as being unpatentable over DiNicola, Ross and Nagata and further in view of Sun Microsystems ("Introducing Swing"; hereinafter referred to as "Sun").

Claims 14-17 are rejected under 35 U.S.C. §103(a) as being unpatentable over DiNicola, Ross, Nagata, Fowler and Sun.

The rejections of claims 1-22 are being appealed.

STATUS OF AMENDMENTS

All amendments to the claims have been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

In general, the present invention relates to techniques for the enabling and disabling of dynamic image buffering by a platform independent application program interface. Each claim being appealed is summarized below. References to the Specification refer to the paragraphs of the published application, U.S. Pat. Pub. No. 2002/0180793, published December 5, 2002.

Claim 1 is an independent apparatus claim that relates to a display system (**10**, **FIG. 1**; **¶[0058]**). **Claim 1** includes a display (**16**, **FIG. 1**; **¶[0058]**), a display buffer (**FIG. 14**; **¶[0131]**) coupled to the display (**16**, **FIG. 1**; **¶[0058]**), and a processor (**12**, **FIG. 1**; **¶[0058]**) adapted to execute an application program (**28**, **FIG. 2**; **¶[0061]**) which, when executed, produces images (**26**, **FIG. 2**; **¶[0063]**) upon the display (**16**, **FIG. 1**; **¶[0058]**), wherein during a first mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**) the images are forwarded in sequence to the display (**16**, **FIG. 1**; **¶[0058]**), and wherein during a second mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**) the images are compiled as a combination image (**228**, **230**, **232** and **234**, **FIG. 14**; **¶[0131]**) of at least one of said images drawn over at least another of said images and presented to the buffer before being forwarded to the display (**16**, **FIG. 1**; **¶[0058]**).

Claim 2 is a dependent apparatus claim that includes all the elements of **Claim 1**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 2** includes the limitations that the application program (**28**, **FIG. 2**; **¶[0061]**) disables or enables buffering (**252**, **FIG. 15**; **¶[0133]**) of the images (**228**, **230**, **232** and **234**, **FIG. 14**; **¶[0131]**) by configuring the processor (**12**, **FIG. 1**; **¶[0058]**) to execute in either the first or second mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**).

Claim 3 is a dependent apparatus claim that includes all the elements of **Claim 1**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 3** includes the limitations that the images (**228**, **230**, **232** and **234**, **FIG. 14**; **¶[0131]**) comprise frame (**228**, **FIG. 14**; **¶[0131]**), panel (**230**, **FIG. 14**; **¶[0131]**) and button (**232** and **234**, **FIG. 14**; **¶[0131]**) images.

Claim 4 is a dependent apparatus claim that includes all the elements of **Claim 1**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 4** includes the limitations that the application program (**28**, **FIG. 2**; **¶[0061]**) comprises a Java program (**¶[0064]**).

Claim 5 is an independent method of manufacture claim that relates to a display system (**10**, **FIG. 1**; **¶[0058]**). **Claim 5** includes a computer-readable storage medium (**18**, **FIG. 1**; **¶[0058]**), an operating system (**40**, **FIG. 2**; **¶[0061]**), an application program (**28**, **FIG. 2**; **¶[0061]**) running on code compatible with the operating system; and a software component (**22**, **FIG. 2**; **¶[0061]**) invoked by the application program (**28**, **FIG. 2**; **¶[0061]**) to display object code (**24**, **FIG. 2**; **¶[0061]**) which, when executed, produces a sequence of images (**10**, **FIG. 1**; **¶[0058]**) upon a display screen (**10**, **FIG. 1**; **¶[0058]**), wherein the software component (**22**, **FIG. 2**; **¶[0061]**) can be configured during runtime of the application program (**28**, **FIG. 2**; **¶[0061]**) to enable or disable buffering (**252**, **FIG. 15**; **¶[0133]**) of the sequence of images as a combination image (**228**, **230**, **232** and **234**, **FIG. 14**; **¶[0131]**) before sending the combination image to the display (**16**, **FIG. 1**; **¶[0058]**).

Claim 6 is a dependent method of manufacture claim that includes all the elements of **Claim 5**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 6** includes the limitations that the object code (**24**, **FIG. 2**; **¶[0061]**) is part of a graphical user interface (**¶[0058]**) associated with the application program (**28**, **FIG. 2**; **¶[0061]**).

Claim 7 is a dependent method of manufacture claim that includes all the elements of **Claim 5**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 7** includes the limitations that the software component (**22**, **FIG. 2**; **¶[0061]**) comprises an application program interface (**22**, **FIG. 2**; **¶[0061]**) of code that translates between code within the application program (**28**, **FIG. 2**; **¶[0061]**) and the operating system (**40**, **FIG. 2**; **¶[0061]**).

Claim 8 is a dependent method of manufacture claim that includes all the elements of **Claims 5** and **7**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 8** includes the limitations that a behavior of the application program interface (**22**, **FIG. 2**; **¶[0061]**) emulates that of a second application program interface (**20**, **FIG. 2**; **¶[0059]**) based on windowing protocols (**¶[0059]-[0060]**) of a windows-based version of said operating system (**40**, **FIG. 2**; **¶[0061]**).

Claim 9 is a dependent method of manufacture claim that includes all the elements of **Claims 5**, **7** and **8**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 9** includes the limitations that the second application program interface (**20**, **FIG. 2**; **¶[0059]**) comprises a Java abstract windowing toolkit (**250**, **FIG. 15**; **¶[0133]**).

Claim 10 is a dependent method of manufacture claim that includes all the elements of **Claim 5**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 10** includes the limitations that the application program (**22**, **FIG. 2**; **¶[0061]**) is written in Java programming language (**¶[0064]**).

Claim 11 is a dependent method of manufacture claim that includes all the elements of **Claim 5**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 11** includes the limitations that the operating system (**40**, **FIG. 2**; **¶[0061]**) is a Windows (**¶[0064]**), Unix (**¶[0064]**) or OS/2 (**¶[0138]**) computer operating system .

Claim 12 is an independent method claim for displaying images (**26**, **FIG. 2**; **¶[0063]**) upon a display (**16**, **FIG. 1**; **¶[0058]**). **Claim 12** includes providing an application program (**28**, **FIG. 2**; **¶[0061]**) running under an operating system (**40**, **FIG. 2**; **¶[0061]**), creating the images of the objects using an interface (**22**, **FIG. 2**; **¶[0061]**) independent of the operating system (**40**, **FIG. 2**; **¶[0061]**), enabling or disabling buffering (**252**, **FIG. 15**; **¶[0133]**) of the images during runtime as directed by the application program (**28**, **FIG. 2**; **¶[0061]**), and wherein during a first mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**), said buffering is disabled so that the images are forwarded in sequence to the display; and during a second mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**), said buffering is enabled so that the images are compiled as a combination image (**228**, **230**, **232** and **234**, **FIG. 14**;

¶[0131]) of at least one of said images drawn over at least another of said images and presented to the buffer (**FIG. 14**; ¶[0131]) before being forwarded to the display (**16**, **FIG. 1**; ¶[0058]).

Claim 13 is a dependent method claim that includes all the elements of **Claim 12**, as described above, for displaying images (**26**, **FIG. 2**; ¶[0063]) upon a display (**16**, **FIG. 1**; ¶[0058]). In addition, **Claim 13** includes the limitations that the creating comprises compiling the object as code that includes part of a graphical user interface (¶[0058]) associated with the application program (**28**, **FIG. 2**; ¶[0061]).

Claim 14 is a dependent method claim that includes all the elements of **Claim 12**, as described above, for displaying images (**26**, **FIG. 2**; ¶[0063]) upon a display (**16**, **FIG. 1**; ¶[0058]). In addition, **Claim 14** includes the limitations that the creating comprises implementing a call routine to compile a software component (**22**, **FIG. 2**; ¶[0061]) that includes an application program interface (**22**, **FIG. 2**; ¶[0061]) between the application program (**28**, **FIG. 2**; ¶[0061]) and the operating system (**40**, **FIG. 2**; ¶[0061]).

Claim 15 is a dependent method claim that includes all the elements of **Claims 12** and **14**, as described above, for displaying images (**26**, **FIG. 2**; ¶[0063]) upon a display (**16**, **FIG. 1**; ¶[0058]). In addition, **Claim 15** includes the limitations that a behavior of the application program interface (**22**, **FIG. 2**; ¶[0061]) emulates that of a second application program interface (**20**, **FIG. 2**; ¶[0059]) based on the operating system (**40**, **FIG. 2**; ¶[0061]).

Claim 16 is a dependent method claim that includes all the elements of **Claims 12**, **14** and **15**, as described above, for displaying images (**26**, **FIG. 2**; ¶[0063]) upon a display (**16**, **FIG. 1**; ¶[0058]). In addition, **Claim 16** includes the limitations that the second application program interface (**20**, **FIG. 2**; ¶[0059]) comprises a Java abstract windowing toolkit (**250**, **FIG. 15**; ¶[0133]).

Claim 17 is a dependent method claim that includes all the elements of **Claim 12**, as described above, for displaying images (**26**, **FIG. 2**; **¶[0063]**) upon a display (**16**, **FIG. 1**; **¶[0058]**). In addition, **Claim 17** includes the limitations that the application program (**28**, **FIG. 2**; **¶[0061]**) is written in Java programming language (**¶[0064]**).

Claim 18 is a dependent method claim that includes all the elements of **Claim 12**, as described above, for displaying images (**26**, **FIG. 2**; **¶[0063]**) upon a display (**16**, **FIG. 1**; **¶[0058]**). In addition, **Claim 18** includes the limitations that the operating system comprises a Windows (**¶[0064]**), Unix (**¶[0064]**) or OS/2 (**¶[0138]**) computer operating system.

Claim 19 is an independent claim that relates to a computer-readable storage device (**18**, **FIG. 1**; **¶[0058]**). **Claim 19** includes a windows-based operating system (**40**, **FIG. 2**; **¶[0061]**), an application program (**28**, **FIG. 2**; **¶[0061]**) running under the operating system (**40**, **FIG. 2**; **¶[0061]**), a plurality of objects (**24**, **FIG. 2**; **¶[0061]**) created at runtime by the application program (**28**, **FIG. 2**; **¶[0061]**), an interface (**22**, **FIG. 2**; **¶[0061]**) independent of the operating system (**40**, **FIG. 2**; **¶[0061]**) adapted for creating images (**26**, **FIG. 2**; **¶[0063]**) of the objects (**24**, **FIG. 2**; **¶[0061]**) and enabling or disabling buffering (**252**, **FIG. 15**; **¶[0133]**) of the images (**26**, **FIG. 2**; **¶[0063]**) to a memory storage area (**FIG. 14**; **¶[0131]**) prior to displaying the images, as directed by the application program (**28**, **FIG. 2**; **¶[0061]**), wherein, during a first mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**), the buffering is disabled so that the images are forwarded in sequence to a display and, during a second mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**), said buffering is enabled so that the images are compiled as a combination image (**228**, **230**, **232** and **234**, **FIG. 14**; **¶[0131]**) of at least one of said images drawn over at least another of said images and presented to the memory storage area (**FIG. 14**; **¶[0131]**) before being forwarded to the display (**16**, **FIG. 1**; **¶[0058]**).

Claim 20 is a dependent apparatus claim that includes all the elements of **Claims 1** and **2**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 20** includes the limitations that the processor (**12**, **FIG. 1**; **¶[0058]**) executes in the first mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**) when the display (**16**, **FIG. 1**; **¶[0058]**) is directly coupled to the processor (**12**, **FIG. 1**; **¶[0058]**).

Claim 21 is a dependent apparatus claim that includes all the elements of **Claims 1** and **2**, as described above, to a display system (**10**, **FIG. 1**; **¶[0058]**). In addition, **Claim 21** includes the limitations that the processor (**12**, **FIG. 1**; **¶[0058]**) executes in the second mode (**¶¶[00133]**, **[0137]**, **[0143]** and **[0145]**) when the display (**16**, **FIG. 1**; **¶[0058]**) is remotely coupled to the processor (**12**, **FIG. 1**; **¶[0058]**).

Claim 22 is a dependent claim that includes all the elements of **Claim 19**, as described above, to a computer-readable storage device (**18**, **FIG. 1**; **¶[0058]**). **Claim 22** also includes a software component (**250**, **FIG. 15**; **¶[0133]**) associated with the interface and adapted to: determine whether the application program is operating in a remote or direct mode, and create a peer component (**252**, **FIG. 15**; **¶[0133]**) to enable or disable buffering of the graphical representation (**26**, **FIG. 2**; **¶[0063]**) of the object based on the determination made by the software component.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether a group of claims consisting of **independent claims 1 and 5 and dependent claims 2, 6, 20 and 21** are unpatentable over DiNicola et al. (U.S. Pat. No. 4,951,229; hereinafter referred to as "DiNicola") in view of Ross (U.S. Pat. No. 5,8838,336) and further in view of Nagata (U.S. Pat. No. 6,522,341).

2. Whether a group of claims consisting of **dependent claims 3, 4 and 7-10** are unpatentable under 35 U.S.C. §103(a) over DiNicola, Ross and Nagata and further in view of Fowler ("Mixing Heavy and Light Components").

3. Whether a group of claims consisting of **independent claims 12 and 19 and dependent claims 11, 13, 18 and 22** are unpatentable over DiNicola, Ross and Nagata and further in view of Sun Microsystems ("Introducing Swing"; hereinafter referred to as "Sun").

4. Whether a group of claims consisting of **dependent claims 14-17** are unpatentable under 35 U.S.C. §103(a) over DiNicola, Ross, Nagata, Fowler and Sun.

ARGUMENTS OF APPELLANTS

I. **Whether a group of claims consisting of independent claims 1 and 5 and dependent claims 2, 6, 20 and 21 are unpatentable over DiNicola et al. (U.S. Pat. No. 4,951,229; hereinafter referred to as "DiNicola") in view of Ross (U.S. Pat. No. 5,8838,336) and further in view of Nagata (U.S. Pat. No. 6,522,341).**

Claims 1 and 5

With respect to independent claims 1 and 5, the Final Office Action, dated October 5, 2009, (F.O.A.), in a Response to Arguments section, states:

> DiNicola teaches in column 2, lines 25-39 and in column 2, line 66 through column 3, line 13, displaying images separately on a display or combing them into a composite image **in the frame buffer prior to displaying**. This teaching is further supplemented by Nagata who further teaches the steps of a mixer mixing two images and then transmitting the combined image to a buffer prior to displaying the image (see column 2, line 15 through column 3, line 4 and in figures 1 and 2).

(p. 22, line 21 through p. 23, line 6; *emphasis added*). However, in a BPAI decision (No. 2007-3520; hereinafter referred to as the "BPAI Decision") previously decided in the current application, the BPAI Decision explicitly states, "From our review of the teachings of the DiNicola reference, we agree with Appellant that **DiNicola does not expressly teach nor inherently buffer the mixed image before forwarding the mixed image to the display**." (p. 8, lines 3-5; *emphasis added*). The F.O.A. also concedes this point by stating, "[DiNicola and Ross] don't specifically teach combining the two images into one image and buffering this combined image prior to display." (p. 4, lines 17-19). Rather Nagata is relied upon for this element.

However, Nagata does not disclose this feature because the images of Nagata are does not actually combine the image in the buffer. Nagata describes the process as follows:

> The image memory for us in image mixing may have a storage capacity equivalent to one line of a raster-scan display device, for

> example. In such a case, the image fraction has a size of one line, and a frame, which is made up of the combined lines that have been sequentially output for the image memory, is presented on the display device.

(col. 1, lines 59-67). In other words, Nagata combines and then stores raster lines rather than storing and outputting whole combined images from the buffer.

The F.O.A. also states, "[I]mage elements of a image are images themselves (see column 2, lines 25-39 and in column 2, line 66 through column 3, line 13)." (p. 23, lines 14-16). Although this may be true in some circumstances, Appellants submit that neither the "raster lines" of Nagata nor the "bit planes" of DiNicola may be analogized as images. DiNicola concedes as much by stating, "None of the planes represent the full structure of the image; it is only the combination of the several planes that allows the final image to be made apparent." (col. 4, lines 41-44). Further, the logic of the BAPI Decision, referenced above, would seem to support this analysis.

The F.O.A. relies upon DiNicola for the element of a mode in which "images are forwarded in sequence to the display" and a mode in which "images are compiled as a combination of image of at least one of said images." However, DiNicola is misconstrued as "forwarding images in sequence to the display. Specifically, the cited portion of DiNicola describes a system that can be "configured to display the resulting independent images separately on the display monitor." This excerpt does not describe the manner, either sequentially or combined, in which the images are "forwarded to the display" but rather merely that which ultimately ends up on the display. Considering the fact that images are dispersed throughout bit planes of DiNicola'a buffer, as shown **FIG. 2** and **3**, it is unlikely the images are ever separated again prior to display. Further, DiNicola shows an image mixer such as image mixer **32** of **FIG. 1** processing the bit planes prior to display. Certainly the images are not separated again one processed by mixer **32**.

To further illustrates that DiNicola does not suggest a system in which images are forwarded in sequence to the display, DiNicola's bit planes 1, 2, 3 ... n **24**, **26**, **28** and **30** are not "buffers" storing "images" in the any sense that would enable them to

implement that aspect of the claimed subject matter. DiNicola states:

> FIG. 2 and FIG. 3 illustrate the differences between bit encoding using bit plane and lateral bit encoding of picture element data. FIG. 2 illustrates three bit planes 70, 72, and 74. A picture element (pel) corresponding to a given location on the screen is represented by a single bit FIG. 2 the first pel is represented by bits c.sub.0, c.sub.1, c.sub.2. The next pel of the display image would be represented in the next bit position in each plane, namely d.sub.0, d.sub.1, and d.sub.2. **The information stored in any one plane represents only a subset of the information required to create the picture element on the display monitor. None of the planes represent the full structure of the image; it is only the combination of the several planes that allows the final image to be made apparent**.

(col. 4, lines 31-44; *emphasis added*). In other words, even if bit planes **24**, **26**, **28**, **30**, **70**, **72** and **74**, are sent to mixer **32**, either sequentially or as a composite, the technology is irrelevant because DiNicola's bit planes, as described above, do not represent "images" but rather merely elements of images.

Further, the storage techniques employed by DiNicola do not enable "one of said images to [to be] drawn over at least another one of said images" because DiNicola stores images as composites in which a single image is spread among multiple bit planes. In other words, when DiNicola combines bit planes, each bit plane may represent multiple images, which cannot be construed as drawing one image over another and, as explained above, the output of the bit planes end up at mixer **32** rather than a display.

With respect to combining DiNicola with other art to provide the "buffering the composite image prior to display," DiNicola specifically teaches away from this element by claiming that the system provides the benefit of not doing so. The following excerpt illustrates this point:

> A hardware implementation of image mixing allows the images to be combined and written directly to the video display monitor without

> generating an intermediate frame buffer containing the composite image. This technique improves display system efficiency in computer devices with limited processing power because movement of an object in one plane, or changes to the images in any plane, does not require a complete regeneration of an intermediate frame buffer.

(col. 2, line 66 through col. 3, line 6). Therefore, it would be improper to combine DiNicola with other art that describes buffering the composite image.

The F.O.A. concedes that DiNicola does not teach a "second mode buffering the combination image prior to display" (p.3, lines 23-24) but instead relies upon Ross for this particular element. However, the "hardware cursor" and "overlay" modes of Ross are not analogous to Applicants' first and second mode because, in Ross, the overlay mode represents the display of alternative images rather than a composite image. Specifically, Ross provides a system in which a second image can be prepared to replace a first image while the first image is being displayed. Obviously, Ross does is not suggesting "buffering the combination prior to display." In addition, merely providing a reference to a system in which "switching" may occur is not enough to suggest Applicants' specific type of shifting. In other words, shifting is not enough and the cited art does not show a shifting between the particular modes described in Applicants' claimed subject matter.

The F.O.A. also concedes that DiNicola and Ross do not teach a "combining the two images into one image and buffering the combined image prior to display" (p.4, lines 15-16) but instead relies upon Nagata for as "mixing the two images and then transmitting the combines image to a buffer prior to displaying the image" (p.4, lines 19-19). However, Nagada does not transmit the combined image to a buffer prior to displaying the image. The only thing shown in Nagada is the mixing and the sending from the mixer. In other words, there is no "transmission to a buffer" after the combining and prior to the displaying.

Therefore, Appellants submit that the current grounds of rejection are in error and that the independent claims 1 and 5 are in condition for allowance and respectfully request reversal of the §103a) rejections.

Claims 2, 6, 20 and 21

Claims 2, 6, 20 and 21, which are each dependent upon one of claims 1 or 5, are allowable for the same reasons described above with respect to claims 1 and 5, specifically that DiNicola, Ross and Nagata, either alone or in combination, teach or suggest Appellants' claimed subject matter. In addition, each of claims 2, 6, 20 and 21 are allowable as being dependent upon an allowable base claim. Therefore, Appellants submit that the current grounds of rejection are in error and that dependent claims 2, 6, 20 and 21 are in condition for allowance and respectfully request reversal of the §103a) rejections.

2. Whether a group of claims consisting of dependent claims 3, 4 and 7-10 are unpatentable under 35 U.S.C. §103(a) over DiNicola, Ross and Nagata and further in view of Fowler ("Mixing Heavy and Light Components").

Claims 3, 4 and 7-10

Claims 3, 4 and 7-10, which are each dependent upon one of claims 1 or 5, are allowable for the same reasons described above with respect to claims 1 and 5. Fowler is added merely to disclose the elements of "frame." "panel" and "button" images and "Mixing Swing in and AWT in the same application program." (p. 9, lines 12-15). Fowler does not disclose any elements lacking in DiNicola, Ross and Nagata. Thus, DiNicola, Ross, Nagata and Fowler, alone or in combination, neither teach nor suggest Appellants' claimed subject matter. In addition, each of claims 3, 4 and 7-10 are allowable as being dependent upon an allowable base claim. Therefore, Appellants submit that the current grounds of rejection are in error and that dependent claims 3, 4 and 7-10 are in condition for allowance and respectfully request reversal of the §103a) rejections.

3. Whether a group of claims consisting of **independent claims 12 and 19 and dependent claims 11, 13, 18 and 22** are unpatentable over DiNicola, Ross and Nagata and further in view of Sun Microsystems ("Introducing Swing"; hereinafter referred to as "Sun").

Claims 12 and 19

Independent claims 12 and 19, which include similar elements of claims 1 or 5, are allowable for the same reasons described above with respect to claims 1 and 5. Sun is added merely to disclose the elements of "Windows, Unix, or OS/2 computer operating system (p. 13, lines 14-15). Sun does not disclose any elements lacking in DiNicola, Ross and Nagata. Thus, DiNicola, Ross, Nagata and Sun, alone or in combination, neither teach nor suggest Appellants' claimed subject matter. Therefore, Appellants submit that the current grounds of rejection are in error and that dependent claims 12 and 19 are in condition for allowance and respectfully request reversal of the §103a) rejections.

Claim 11

Claim 11, which is dependent upon claim 5, is therefore allowable for the same reasons described above with respect to claims 1 and 5. Sun is added merely to disclose the elements of "Windows, Unix, or OS/2 computer operating system (p. 13, lines 14-15). Sun does not disclose any elements lacking in DiNicola, Ross and Nagata. Thus, DiNicola, Ross, Nagata and Sun, alone or in combination, neither teach nor suggest Appellants' claimed subject matter. In addition, claim is allowable as being dependent upon an allowable claim 5. Therefore, Appellants submit that the current grounds of rejection are in error and that dependent claim 11 is in condition for allowance and respectfully request reversal of the §103a) rejection.

Claims 13, 18 and 22

Claims 13, 18 and 22, each of which depends upon one of claims 12 or 19, are allowable for the same reasons described above with respect to claims 1, 5, 12 and 19. Sun is added merely to disclose the elements of "Windows, Unix, or OS/2 computer operating system (p. 13, lines 14-15). Sun does not disclose any elements lacking in DiNicola, Ross and Nagata. Thus, DiNicola, Ross, Nagata and Sun, alone or in combination, neither teach nor suggest Appellants' claimed subject matter. In addition, each claim is allowable as being dependent upon one of an allowable base claim. Therefore, Appellants submit that the current grounds of rejection are in error and that dependent claims 13, 18 and 22 are in condition for allowance and respectfully request reversal of the §103a) rejection.

4. Whether a group of claims consisting of dependent claims 14-17 are unpatentable under 35 U.S.C. §103(a) over DiNicola, Ross, Nagata, Fowler and Sun.

Claims 14-17

Claims 14-17 each of which depends upon claim 12, are allowable for the same reasons described above with respect to claims 1, 5, 12 and 19. Sun is added merely to disclose the elements of "Windows, Unix, or OS/2 computer operating system (p. 13, lines 14-15). Neither Fowler nor Sun disclose any elements lacking in DiNicola, Ross and Nagata. Fowler is added merely to disclose the elements of "frame." "panel" and "button" images and "Mixing Swing in and AWT in the same application program." (p. 9, lines 12-15). Sun does not disclose any elements lacking in DiNicola, Ross and Nagata. Thus, DiNicola, Ross, Nagata, Fowler and Sun, alone or in combination, neither teach nor suggest Appellants' claimed subject matter. In addition, each claim is allowable as being dependent upon one of an allowable base claim. Therefore, Appellants submit that the current grounds of rejection are in error and that dependent claims 14-17 are in condition for allowance and respectfully request reversal of the §103a) rejection.

CONCLUSION

In addition, the combination of DiNicola, Ross, Nagada, Fowler and Sun do not suggest, either alone or in combination, the specific of Applicants' switching feature because none of these references illustrate both of the modes between which the Applicants' system is switched. To establish *prima facie* obviousness of a claimed invention under §103(a), all the claim limitations must be taught or suggested by the prior art. (M.P.E.P., §2143,03, citing *in re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (CCPA 1974)). In addition, "**All words in a claim must be considered** in judging the patentability of that claim against prior art." (*Id.*, citing *In re Wilson*, 424 F.2d 1382, 1385; 165 U.S.P.Q. 494, 496 (CCPA 1970); *emphasis added*). Applicants believe that the cited art fails to meet this standard.

Therefore, Appellants submit that the current grounds of rejection are in error and that the independent claims 1, 5, 12 and 19 and dependent claims 2-4, 6-11, 13-18 and 20-22 are in condition for allowance. In addition dependent claims 2-4, 6-11, 13-18 and 20-22 are allowable because they each depend upon one of the allowable independent claims. Therefore, a reversal of the §103(a) rejections of claims 1-22 is respectfully solicited.

Respectfully submitted,

Date: March 5, 2010

/Gregory K. Goshorn/
By: Gregory K. Goshorn
Reg. No.: 44,721
ATTORNEY FOR APPELLANTS

Greg Goshorn, P.C.
9600 Escarpment
Suite 745-9
Austin, Texas 78749
Telephone: (512) 291-9203
Facsimile: (512) 535-4206

CLAIMS APPENDIX

(Currently Pending Claims)

1. (Previously presented) A display system, comprising:

a display;

a display buffer coupled to the display; and

a processor adapted to execute an application program which, when executed, produces images upon the display, wherein during a first mode the images are forwarded in sequence to the display, and wherein during a second mode the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the buffer before being forwarded to the display.

2. (Previously presented) The display system as recited in claim 1, wherein the application program disables or enables buffering of the images by configuring the processor to execute in either the first or second mode.

3. (Original) The display system as recited in claim 1, wherein the images comprise frame, panel and button images.

4. (Original) The display system as recited in claim 1, wherein the application program comprises a Java program.

5. (Previously presented) A computer-readable storage medium, comprising:

an operating system,

an application program running on code compatible with the operating system; and

a software component invoked by the application program to display object code which, when executed, produces a sequence of images upon a display screen, wherein the software component can be configured during runtime of the application program to enable or disable buffering of the sequence of images as a combination image before sending the combination image to the display.

6. (Previously presented) The medium as recited in claim 5, wherein the object code is part of a graphical user interface associated with the application program.

7. (Previously presented) The medium as recited in claim 5, wherein the software component comprises an application program interface of code which translates between code within the application program and the operating system.

8. (Previously presented) The medium as recited in claim 7, wherein a behavior of the application program interface emulates that of a second application program interface based on windowing protocols of a windows-based version of said operating system.

9. (Previously presented) The medium as recited in claim 8, wherein the second application program interface comprises a Java abstract windowing toolkit.

10. (Previously presented) The medium as recited in claim 5, wherein the application program is written in Java programming language.

11. (Previously presented) The medium as recited in claim 5, wherein the operating system comprises a Windows, Unix or OS/2 computer operating system.

12. (Previously presented) A method for displaying images of objects upon a display, the method comprising:

providing an application program running under an operating system;

creating the images of the objects using an interface independent of the operating system; and

enabling or disabling buffering of said images during runtime as directed by the application program, wherein:

during a first mode, said buffering is disabled so that the images are forwarded in sequence to the display; and

during a second mode, said buffering is enabled so that the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the buffer before being forwarded to the display.

13. (Original) The method as recited in claim 12, wherein said creating comprises compiling the object as code that includes part of a graphical user interface associated with the application program.

14. (Original) The method as recited in claim 12, wherein said creating comprises implementing a call routine to compile a software component that includes an application program interface between the application program and the operating system.

15. (Previously presented) The method as recited in claim 14, wherein a behavior of the application program interface emulates that of a second application program interface based on the operating system.

16. (Previously presented) The method as recited in claim 15, wherein the second application program interface comprises a Java abstract windowing toolkit.

17. (Original) The method as recited in claim 12, wherein the application program is written in Java programming language.

18. (Original) The method as recited in claim 12, wherein the operating system comprises a Windows, Unix or OS/2 computer operating system.

19. (Previously presented) A computer-readable storage device, comprising:

a windows-based operating system;

an application program running under the operating system;

a plurality of objects created at runtime by the application program;

an interface independent of the operating system, and adapted for:

creating images of the objects; and

enabling or disabling buffering of the images to a memory storage area prior to displaying the images, as directed by the application program, wherein:

during a first mode, said buffering is disabled so that the images are forwarded in sequence to a display; and

during a second mode, said buffering is enabled so that the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the memory storage area before being forwarded to the display.

20. (Previously presented) The display system as recited in claim 2, wherein the processor executes in the first mode when the display is directly coupled to the processor.

21. (Previously presented) The display system as recited in claim 2, wherein the processor executes in the second mode when the display is remotely coupled to the processor.

22. (Previously presented) The computer-readable storage device as recited in claim 19, further comprising a software component associated with the interface and adapted to:

(i) determine whether the application program is operating in a remote or direct mode, and

(ii) create a peer component to enable or disable buffering of the graphical representation of the object based on the determination made by the software component.

EVIDENCE APPENDIX

No evidence has been submitted in conjunction with this application.

RELATED PROCEEDINGS APPENDIX

There are currently no related proceedings associated with this application.